

유비쿼터스 환경을 고려한 시맨틱 웹 데이터 관리 기법 연구

김연희*, 김지현**

요약

미래 네트워크 모델로 주목받고 있는 유비쿼터스 환경의 보편화를 위해서 사용자 중심의 지능화된 유비쿼터스 컴퓨팅 기반 시스템 개발이 요구된다. 특히, 동적으로 변화하는 상황 정보를 인지하고 사용자의 다양한 요구에 적합한 서비스를 추론하여 제공하기 위해서는 사용자와 장치들간의 지식 공유와 명확한 의사 소통이 가능해야 한다. 본 논문은 의미적 연관성과 추론에 기반한 사용자와 장치들간의 상호작용을 제공하기 위해서 각 장치들이 제공하는 서비스나 데이터의 의미를 온톨로지와 메타데이터를 이용하여 기술하고 효율적으로 관리하는데 목표를 두고 있다. 본 논문에서는 시맨틱 데이터를 OWL로 기술하여 다양한 의미적 관계를 표현하고 이것을 활용하는 지능적 유비쿼터스 기반 시스템을 제안한다. 그리고 각 서브 시스템별로 시맨틱 데이터를 관리하기 위한 저장 및 인덱스 구조와 질의 처리 전략을 제안하고 이를 지원하기 위해 시맨틱 데이터의 주요 요소인 클래스와 리소스를 식별하는 레이블링 기법을 사용한다. 본 논문에서 제안한 유비쿼터스 시스템의 시맨틱 데이터 관리 기법을 통해 보다 정확하고 사용자의 요구를 폭넓게 만족시키는 장치를 추론에 의해 검색할 수 있다.

Schemes for Managing Semantic Web Data in Ubiquitous Environment

Youn Hee Kim*, JeeHyun Kim**

Abstract

One important issue to generalize the ubiquitous paradigm is the development of user-centralized and intelligent ubiquitous computing systems. Sharing knowledge and correct communication between users and devices are needed to be aware of continuous changed context information and infer services for which users are suited. The goal of this paper is to describe and manage effectively the meaning of services or data which each device offers for interaction between users and devices based on semantic relationships and reasoning. In this paper, we represent semantic data using OWL and design a ubiquitous based intelligent system. We propose some index structures and strategies to process queries classified by each subsystem and adopt labeling schemes to identify classes and resources in the semantic data. We can find devices which satisfies various user's requests exactly and quickly using the proposed strategies.

Keywords : Ubiquitous Environment, Ontology, Metadata, OWL, Index, Query Processing

1. 서론

유비쿼터스 컴퓨팅은 다양한 장치들이 보이지

않는 네트워크로 연결되어 있으면서 사용자들이 장소나 시간에 관계없이 자유롭게 장치에 접근하여 적절한 서비스를 제공받을 수 있는 환경을 의미한다[1]. 유비쿼터스 컴퓨팅 환경은 언제 어디서나 어떠한 장치를 통해서든 컴퓨팅이 가능하도록 하는 것을 목표로 하는 최근 연구 이슈로서 장치 내부에 장착되는 칩이나 장치의 지능화 및 소형화 기술, 광대역 통신과 컨버전스 기술의 일반화, 네트워크 관련 기술 등에 대한 연구가 중요하게 평가되어 왔다[2]. 그러나 이러한 기술은 정보 통신 분야에 치우친 연구 내용으로

※ 제일저자(First Author) : 김연희
접수일:2008년 12월 01일, 완료일:2009년 02월 27일
* 부천대학 e-비즈니스과
ellyblue06@gmail.com
** 서일대학 소프트웨어과
▣ 본 논문은 2007년도 서일대학 학술연구비에 의해 연구되었음.

같은 상황이라도 사용자에게 따라 다른 서비스를 제공할 수 있는 사용자 중심의 지능화된 유비쿼터스 컴퓨팅 기반 시스템 개발을 위해서는 추가적인 요소 기술이 요구된다. 특히, 동적으로 변화하는 유비쿼터스 환경의 상황 정보를 인지하고 각 사용자의 요구에 적합한 서비스를 추론하여 제공하기 위해서는 사용자와 장치들간의 지식 공유 및 명확한 의사 소통이 가능해야 한다. 또한, 유비쿼터스 환경에서의 네트워크는 필연적으로 분산된 형태이기 때문에 이러한 분산된 장치에 각각 존재하는 다양한 정보 리소스들에 대한 의미적 통합 관리 기술이 무엇보다 필요하다.

따라서, 본 논문에서는 유비쿼터스 환경에서의 의미적 연관성과 추론에 기반한 사용자와 장치들간의 상호 작용을 제공하는 것을 목표로 이질적이고 분산된 특성을 가지고 있는 유비쿼터스 컴퓨팅 환경에 온톨로지와 메타데이터를 이용하는 시맨틱 웹 기술을 적용하고자 한다.

시맨틱 웹은 사람뿐만 아니라 컴퓨터와 같은 장치가 이해할 수 있는 웹으로 메타데이터와 온톨로지를 통해서 정보 리소스의 개념을 정의하고 자원들 간의 의미적 연관성을 표현함으로써 현재 웹에 비해 보다 지능적인 정보의 검색은 물론 자동화된 서비스의 제공이 가능하다는 특징을 가지고 있다[3]. 메타데이터는 정보 리소스의 개념과 의미적 관계를 기술한 것이고, 이러한 메타데이터를 기술할 때 발생할 수 있는 의미의 모호성과 중첩성의 문제를 해결하기 위한 것이 온톨로지이다. 온톨로지는 메타데이터를 기술할 때 사용되는 용어의 개념과 용어들 간의 관계를 정의한 것으로 정형화된 지식의 공유가 가능하도록 한다. 이러한 메타데이터와 온톨로지를 형식적으로 기술하기 위해 W3C를 중심으로 RDF(Resource Description Framework), RDF 스키마, DAML+OIL, OWL(Web Ontology Language)과 같은 다양한 언어들 소개되어 왔다. 그 중 OWL은 가장 최근에 소개된 언어로 W3C 권고안으로 채택되었고 풍부한 표현력으로 정교한 표현이 가능하면서 분산 환경에 적합한 특징을 가지고 있어 본 논문에서는 각 장치에 저장되어 있는 정보 리소스의 의미를 기술하기 위한 메타데이터 및 온톨로지 표현의 수단으로 활용한다[4].

한편, 유비쿼터스 환경에서는 데이터가 다양한

장치들에 존재하기 때문에, 각각의 장치에 존재하는 데이터를 논리적으로 통합하여 사용자의 요구에 대해 적절한 처리가 가능해야 한다. 무엇보다 이러한 유비쿼터스 환경의 성능을 결정하는 요인 중에 중요한 것이 사용자 요구에 적합한 데이터를 포함하고 있는 각 장치를 빠르게 찾는 것이다. 더욱이 이것은 장치의 수가 많아질수록 전체 시스템의 성능에 중요한 영향을 미친다. 따라서, 시맨틱 웹 기술을 적용한 유비쿼터스 환경을 제공하기 위해서는 메타데이터와 온톨로지와 같은 시맨틱 데이터의 효과적인 저장과 검색을 위해 저장 및 인덱스 구조나 질의 처리 기법과 같은 데이터베이스 관련 분야와의 연계가 필요하다.

본 논문에서는 의미적 연관성과 추론에 기반한 보다 진보된 유비쿼터스 컴퓨팅 환경의 시스템 개발을 위한 새로운 프레임워크를 제시하고 이를 기반으로 하여 사용자의 요구에 적합한 정보 리소스나 서비스의 검색이 가능하도록 시맨틱 웹 데이터의 관리 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존의 시맨틱 데이터 관리 기법을 두 가지 측면에서 분석한다. 3장에서는 시맨틱 데이터의 표현 및 식별 방법을 소개하고 대표적인 질의 유형을 분류한다. 4장에서는 본 논문에서 제안하는 시스템의 전체 구성을 소개하고 각 구성 요소별로 시맨틱 데이터의 관리 기법을 설명한다. 그리고 5장에서 제안한 기법을 이용한 질의 처리 전략을 소개하고 결론을 맺는다.

2. 관련연구

시맨틱 웹 기술을 적용한 유비쿼터스 환경의 애플리케이션 개발을 위해서 다음과 같이 두 가지 측면에서 기존 연구 결과를 분석할 필요가 있다.

2.1 분산 네트워크 기반 시맨틱 데이터 관리

P2P와 같은 일반적인 분산 네트워크 환경에서 네트워크를 구성하거나 사용자의 요구에 적합한 네트워크 노드를 검색하기 위한 라우팅 기술에 메타데이터와 온톨로지를 이용하려는 시도가 있었다[5][6][7]. 넓은 의미에서 유비쿼터스 환경을 분산 네트워크로 분류할 수 있기 때문에

분산 네트워크 환경에서의 시맨틱 데이터 관리 기술에 대한 분석이 필요하다.

참고문헌 [5]에서는 질의 처리 시 모든 Peer에 질의를 전달해야 하는 비효율적인 문제를 해결하고 질의와 관련된 Peer에만 질의를 전달하여 처리할 수 있도록 하기 위해 Super-Peer를 이용하여 네트워크를 구성하고 RDF로 기술된 메타 데이터를 이용하여 사용자 질의에 적합한 Peer를 찾기 위한 라우팅 기법을 제안하였다. 그러나 온톨로지에 정의되어 있는 용어들간의 계층 구조를 고려하지 않고 리소스의 속성에 기반한 질의 유형을 주로 처리하기 때문에 추론에 기반한 지능적인 질의 처리가 어렵다. 그리고 모든 인덱스가 Super-Peer에 존재하기 때문에 질의 처리에 대한 Super-Peer의 부담이 크다.

참고문헌 [6]에서는 온톨로지를 이용하여 네트워크에서 제공할 수 있는 서비스를 분류하고 그 결과에 따라 계층적인 P2P 네트워크를 구성한다. 그리고 사용자가 서비스의 이름을 제시하면 이에 적합한 서버를 찾은 후 해당 서버에 연결되어 있는 서브 네트워크의 Peer들 중에서 사용자에게 적합한 Peer를 선택한다. 이때, 선택된 Peer와 계층 관계를 가지고 있는 다음 Peer를 계속해서 검색할 때마다 서비스의 속성 정보와 온톨로지 정의를 이용하여 추론기를 통해 계층 관계를 유추하기 때문에 유비쿼터스 환경에 적용하기에는 처리 시간이 오래 걸리는 문제가 존재한다.

참고문헌 [7]에서는 분산되어 있는 각 Peer마다 메타데이터의 저장소를 유지하고 Peer들을 중재하는 역할을 담당하는 미디어이터를 통해 특정 정보 리소스를 검색하고자 하는 사용자의 질의를 처리한다. 그러나 메타데이터에 기술된 정보 리소스의 속성 관계에 기반하여 질의 처리에 적합한 계획을 수립하는 기법 연구에 초점을 맞추고 있으며 온톨로지를 통해 정의한 용어들간의 계층 관계에 대해서 고려하지 않기 때문에 추론을 이용한 리소스의 검색이 불가능하다.

분산 네트워크 환경에서 시맨틱 데이터를 관리하는 기존 연구 결과들은 유비쿼터스 환경에서의 네트워크를 구성하는데 적용할 수 있으나 온톨로지가 정의하고 있는 개념들의 계층 관계와 OWL로 표현할 수 있는 보다 다양한 의미적 관계를 고려하지 않는 경우가 대부분이기 때문

에 추론에 기반한 보다 정확하고 풍부한 정보의 검색이 불가능하다.

2.2 유비쿼터스 기반 시맨틱 데이터 관리

유비쿼터스 컴퓨팅과 관련된 기존의 많은 연구들은 상황 인지를 위해 RFID 등을 기반으로 하는 미들웨어를 개발하는데 초점을 맞추고 있지만 최근에는 네트워크를 구성하고 사용자에게 적합한 서비스를 제공하기 위해 시맨틱 데이터를 활용하는 연구 결과들이 소개되고 있다[8,9].

참고문헌 [8]에서는 사용자가 제시한 질의와 가장 유사한 메타데이터를 유지하고 있는 장치를 검색하기 위한 유사도 함수와 라우팅 기법을 제안한다. 모든 장치들은 메타데이터와 해당 메타데이터를 포함하고 있는 장치에 대한 정보를 유지하고 있고 사용자가 질의 처리를 요청한 경우 유사도 함수를 통해 질의의 메타데이터와 가장 유사한 메타데이터를 보유하고 있는 장치를 선택하여 질의를 전달하게 된다. 그러나 메타데이터는 온톨로지에 비해 기술된 내용이 많기 때문에 질의 처리를 위해 경로 상의 모든 장치에서 메타데이터에 대한 유사도 평가가 이루어지는 것은 시간과 비용적인 측면에서 비효율적이다. 그리고 질의의 전달 경로가 길어지는 경우 질의와 관련된 장치를 보다 빠르게 검색할 수 있도록 하는 구성 요소가 필요하다는 문제점을 가지고 있다.

참고문헌 [9]에서는 온톨로지를 이용해서 서비스를 분류하고 서비스의 입력 및 출력에 대한 의미적 계층 관계를 이용하여 사용자에게 적합한 장치를 찾기 위한 매칭 기법을 제안한다. 제안된 매칭 기법은 유비쿼터스 환경에 적합하도록 간단한 형태를 가지고 있지만 입력과 출력이 없는 일반적인 서비스 검색에는 활용하기 어렵다. 그리고 질의 처리를 위한 인덱스나 저장 구조의 활용에 대해서는 고려하지 않는다.

유비쿼터스 환경에서 시맨틱 데이터를 활용하는 기존 연구들은 메타데이터만을 활용하여 사용자에게 적합한 장치를 검색하거나 특정 서비스에 특화된 간단한 온톨로지를 활용하는 경우가 많다. 따라서 지능적인 장치 검색을 제공하기 위해서는 시맨틱 데이터의 적극적인 활용이 필요하다.

3. 시맨틱 데이터의 표현

유비쿼터스 환경에서 사용자가 원하는 데이터를 저장하고 있거나 원하는 서비스에 적합한 장치를 정확하게 검색하기 위해서는 장치가 제공하는 데이터나 서비스의 의미를 별도로 기술하는 온톨로지와 메타데이터가 필요하다. 그리고 유비쿼터스 환경에서 빠른 시간내에 최소의 비용으로 시맨틱 데이터를 처리하기 위해서는 온톨로지 및 메타데이터에 존재하는 다양한 요소들을 효과적으로 식별할 수 있는 레이블링 기법이 필요하다. 본 절에서는 본 논문에서 사용하는 시맨틱 언어와 레이블링 기법에 대해 설명한다.

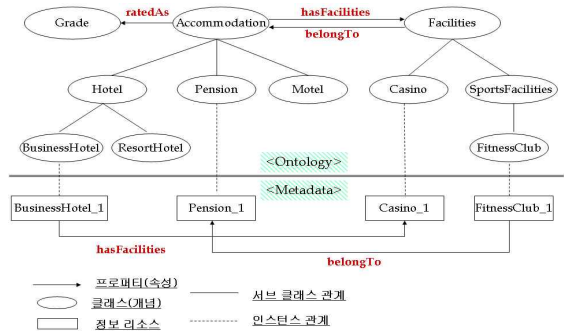
3.1 OWL로 기술된 시맨틱 데이터

OWL은 온톨로지와 메타데이터를 함께 기술할 수 있는 시맨틱 웹 언어로서 RDF/S가 가지는 표현력의 한계를 극복할 수 있는 대안으로 W3C에서 제안되었다[4]. OWL은 개념 및 정보 리소스의 의미를 다양하게 기술할 수 있을 뿐만 아니라 분산되어 있는 개념 및 정보 리소스 간의 의미적 관계를 다양하게 기술할 수 있는 표현력을 제공한다. 따라서 본 논문에서는 유비쿼터스 환경의 특성을 고려하여 시맨틱 데이터의 기술 언어로 OWL을 활용한다.

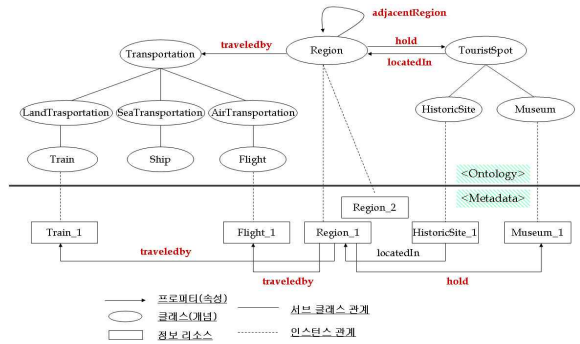
다른 시맨틱 웹 언어들의 특징과 마찬가지로 OWL에서는 온톨로지와 메타데이터가 주어, 서술어, 목적어로 이루어진 트리플 구조의 문장(statement)으로 기술된다. 온톨로지에서는 개념을 정의한 클래스, 메타데이터에서는 특정 클래스의 인스턴스인 정보 리소스가 주어에 해당된다. 서술어는 클래스 및 정보 리소스 간의 의미적 관계인 프로퍼티를 의미하고 목적어는 해당 프로퍼티의 값으로 상황에 따라 리터럴 데이터나 또 다른 클래스 및 정보 리소스가 될 수 있다. OWL로 기술된 시맨틱 데이터는 편리한 처리를 위해 노드와 간선에 모두 레이블이 존재하는 방향성있는 그래프로 표현할 수 있는데 노드는 클래스나 정보 리소스를 표현하고 간선이 프로퍼티를 의미하게 된다.

본 논문에서는 사용자에게 여행과 관련된 서비스를 제공하는 유비쿼터스 환경을 고려한다. 여행과 관련하여 센서 장치들은 숙박 시설과 주

변 관광지에 대한 데이터를 의미와 함께 유지하고 있고 계속적으로 이동하는 사용자가 자신의 장치를 통해 질의 처리를 요청하면 가장 적합한 센서 장치를 추천하여 사용자의 장치가 접속할 수 있도록 하는 것을 목표로 한다.



a. 숙박 시설 온톨로지에 기반한 메타데이터



b. 관광지 온톨로지에 기반한 메타데이터
(그림 1) 시맨틱 데이터의 그래프 모델

(그림 1)은 본 논문에서 가정하고 있는 유비쿼터스 환경에서 센서 장치들이 공유하는 온톨로지와 이에 따라 기술된 메타데이터를 OWL 기반의 그래프 모델로 표현한 것이다. 본 논문에서는 (그림 1)에서 설명하고 있는 숙박 및 관광지와 관련한 두 개의 온톨로지를 사용한다. 숙박 관련 온톨로지는 클래스와 관련하여 3개의 계층 구조를 정의하고 있으며 관광지 관련 온톨로지도 3개의 계층 구조를 포함하고 있다. 특히 본 논문에서는 OWL에서 다양한 의미적 관계 중에서 클래스와 관련한 equivalentClass와 subclassOf의 관계, 프로퍼티와 관련한 inverseOf와 symmetric 관계에 기반한 추론 질의 처리의 효율성을 고려한다.

3.2 시맨틱 데이터를 위한 레이블링 기법

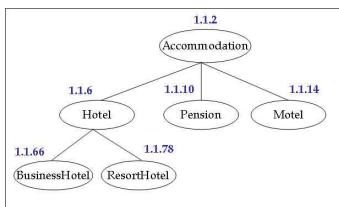
질의 처리 시 온톨로지를 추론하여 얻어지는 새로운 정보까지 질의의 대상으로 처리해야만 보다 정확하고 풍부한 질의 처리가 가능하다. 유비쿼터스 환경에서는 이러한 추론 과정이 실시간으로 이루어지는 것보다는 미리 처리하여 그 결과를 저장한 후 이용하는 것이 비용과 시간적 측면에서 효율적이라 할 수 있다. 그러나 이러한 경우 실제로 질의를 처리할 때 원래의 데이터보다 훨씬 더 많은 데이터를 다루어야 하는 것이 문제가 될 수 있다. 따라서 본 논문에서는 시맨틱 데이터를 구성하는 주요 요소인 클래스와 리소스들을 쉽게 식별하여 데이터의 처리 부담을 줄일 수 있도록 레이블링 기법을 사용한다. 특히, 본 논문에서는 추론 과정에서 가장 중요하게 다루어지는 개념, 즉 클래스 간의 계층 관계를 빠르게 판단하기 위해서 기존에 사용되던 기법 중에서 소수 레이블링 기법을 활용한다.

소수 레이블링은 XML이나 RDF/S에서 나타나는 계층 구조에서 각각의 노드에 고유한 소수 값을 부여받은 후 부모의 소수 레이블 값에 부여받은 소수 값을 곱하여 자신의 레이블로 활용하는 방법이다[10]. 이 방법의 장점은 단순히 두 노드의 소수 레이블 값을 비교하여 나누었을 때 나머지가 0이면 두 노드가 계층 관계를 맺고 있다는 것을 빠르게 확인할 수 있다는 것이다.

본 논문에서는 온톨로지서 정의된 클래스를 식별하기 위해 소수 레이블링을 이용한 다음과 같은 레이블링 기법을 제안한다.

ClassID := (OntologyID.SegmentID.PrimeNum)

OntologyID는 해당 클래스가 정의되어 있는 온톨로지의 식별자를 의미하고, SegmentID는 온톨로지 내에 정의된 계층 구조들 중에서 해당 클래스가 포함되어 있는 계층 구조에 대한 식별자이다. PrimeNum은 해당 클래스에 부여된 소수 레이블이다.



(그림 2) 클래스 레이블링 예

(그림 2)는 (그림 1-a)에서 제시한 숙박 시설 온톨로지에 정의된 일부 클래스들에 레이블을 부여한 결과를 보여준다. 예를 들어, (그림 2)에서 “Hotel” 클래스의 레이블은 1.1.6인데 맨 앞에 1은 첫 번째 온톨로지에 정의되어 있음을 의미하고 두 번째 1의 의미는 첫 번째 온톨로지에 존재하는 3개의 계층 구조 중 첫 번째 계층 구조에 정의되어 있음을 의미한다. 마지막으로 6은 자신에게 부여된 소수값 3에 부모인 “Accommodation” 클래스의 소수 레이블인 2를 곱한 결과로 배정된 자신의 소수 레이블이다.

정보 리소스를 식별하기 위한 레이블은 해당 리소스의 클래스 타입과 리소스 정의 순서를 나타낼 수 있도록 “클래스타입_정의순서” 형태로 부여한다. 예를 들어, (그림 1-a)에서 리소스에 부여된 “BusinessHotel_1” 레이블의 의미는 BusinessHotel 클래스의 첫 번째 인스턴스로 정의되어 있음을 의미한다. 이러한 리소스의 레이블링 방법은 해당 리소스의 클래스 타입을 쉽게 판단할 수 있다는 장점을 가지고 있다.

3.3 질의 유형 분류

시맨틱 웹 데이터가 트리플 구조의 문장 형태로 표현되기 때문에 사용자의 질의 또한 트리플 구조로 표현할 수 있다. 따라서 주어, 서술어, 목적어 중 무엇을 질의 조건으로 제시하느냐에 따라 다양한 질의 유형이 존재한다. 하지만 유비쿼터스 환경에서는 사용자가 자신이 제공받기를 원하는 서비스의 이름이나 검색을 원하는 정보 리소스의 개념, 즉 클래스 타입을 주요 질의 조건으로 제시하고 추가적으로 해당 서비스나 리소스의 특정 프로퍼티에 대한 값을 질의 조건으로 제시하는 것이 일반적이다.

본 논문에서는 이러한 유비쿼터스 환경의 질의 특성을 고려하여 <표 1>과 같이 대표적인 질의 유형을 선별하고 이러한 질의 유형에 대한 효율적인 처리에 목표를 둔다. (표 1)에서 질의는 (주어 서술어 목적어)의 형태로 기술하고 클래스와 리소스의 인스턴스 관계는 클래스:리소스와 같이 표현하였다. 또한 검색하고자 하는 리소스는 “?”로 나타낸다. (표 1)에서 제시한대로 본 논문에서는 사용자가 검색하기를 원하는 정보 리소스 또는 서비스의 클래스 타입이 필수적인 질의 조건이다. 그리고 추가적으로 제공되는 질

의 조건인 프로퍼티에 대해서는 역관계와 대칭 관계에 기반한 추론적 질의 처리를 고려한다. 이것은 OWL이 프로퍼티에 대한 다양한 의미적 관계를 표현할 수 있다는 특징을 반영한 것으로 기존 유비쿼터스 기반 시맨틱 데이터 관리 시스템과의 차별화된 점이다.

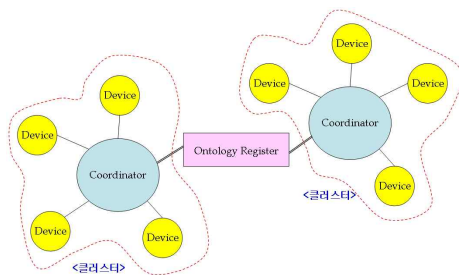
<표 1> 질의 유형 분류

구분	질의 유형
1	<p>정의 트리플 구조에 기반한 일반적인 질의</p> <p>예 (Hotel:? traveledBy Train_1) Train_1 리소스에 의해 도착할 수 있는 Hotel 타입의 리소스를 검색하라.</p>
2	<p>정의 프로퍼티의 역관계(inverseOf) 기반 질의</p> <p>예 (Hotel:? hasFacilities FitnessClub_1) FitnessClub_1 리소스를 편의시설로 가지고 있는 Hotel 타입의 리소스를 검색하라. (hasFacilities 프로퍼티는 belongTo와 역관계 성립)</p>
3	<p>정의 프로퍼티의 대칭관계(symmetric) 기반 질의</p> <p>예 (Region_1 adjacentRegion Region;?) Region_1 리소스에 근접하고 있는 Region 타입의 리소스를 검색하라. (adjacentRegion 프로퍼티는 대칭관계 성립)</p>

4. 시맨틱 데이터의 관리 기법

4.1 시스템 구성

본 논문에서는 사용자 요구에 부합하는 장치를 효율적으로 검색하기 위해 시맨틱 웹 데이터를 활용하는 유비쿼터스 기반 시스템을 제안한다. 본 논문에서 제안한 유비쿼터스 기반 시스템은 (그림 3)과 같이 온톨로지 레지스터(Ontology Register), 코디네이터(Coordinator), 장치(Device)로 구성된 3-레이어의 계층적 형태이다.



(그림 3) 유비쿼터스 기반 시스템 구성

유비쿼터스 환경의 주요 구성 요소인 장치는 크게 정적인 것과 동적인 것으로 분류할 수 있다. 즉, 특정 장소에 정적으로 위치하면서 사용자에게 제공하기 위한 데이터를 주기적으로 수집하는 장치와 사용자가 휴대하는 장치처럼 동적으로 움직이면서 원하는 서비스를 제공받고자 할 때 네트워크에 접속하는 장치를 의미한다.

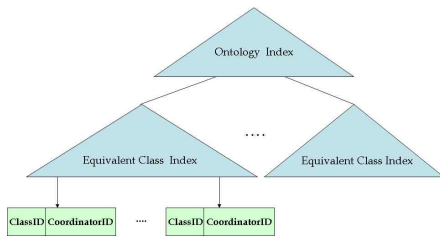
본 논문에서는 사용자에게 제공하기 위한 정보 리소스들 간의 관계, 즉 메타데이터를 저장하고 있는 정적인 장치들을 온톨로지의 유사성에 기반하여 분류하고 논리적인 클러스터를 구성한다. 그리고 나서 각 클러스터마다 공유 온톨로지의 관리와 질의의 분배를 담당하는 코디네이터를 생성한다. 즉, 코디네이터는 동일한 온톨로지의 정의에 따라 작성된 메타데이터 정보를 유지하고 있는 장치들을 논리적으로 모아놓은 클러스터의 관리자라고 할 수 있다. 코디네이터는 논리적 클러스터의 개수만큼 존재하며 특정 클래스 타입의 정보 리소스를 저장하고 있는 장치가 무엇인지에 대한 정보를 유지한다. 온톨로지 레지스터는 각 온톨로지에 정의된 특정 클래스에 대한 정보를 유지하고 있는 코디네이터가 무엇인지에 대한 정보를 유지한다. 온톨로지 레지스터가 질의 요청을 받게 되면 계층적인 시스템 구조를 따라 사용자 장치에 가장 적합한 장치를 검색하게 되고 실제 질의 처리는 해당 장치를 통해 이루어진다. 본 논문에서 (그림 3)과 같이 계층적인 네트워크 형태로 시스템을 구성하는 이유는 사용자가 요청한 질의 처리에 가장 적합한 정적인 장치를 검색하기 위해 필요한 통신 비용을 최대한 줄이면서도 정확한 검색을 가능하도록 하기 위함이다.

4.2 온톨로지 레지스터의 데이터 관리

온톨로지 레지스터는 사용자가 요청한 질의를 처리할 수 있는 장치가 소속되어 있는 코디네이터를 찾아 질의를 전달하는 것이 가장 중요한 역할이다. 본 논문에서는 검색의 대상이 되는 서비스 또는 리소스의 클래스 타입을 주요 질의 조건으로 고려하기 때문에 온톨로지 레지스터에는 특정 클래스 타입에 대한 처리가 가능한 코디네이터를 빠르게 추천해 줄 수 있는 인덱스 구조가 필요하다.

따라서 본 논문에서는 온톨로지 레지스터를

위해 (그림 4)와 같이 온톨로지 인덱스와 클래스 인덱스로 구성된 레지스터 인덱스를 제안한다. (그림 4)에서 온톨로지 인덱스는 사용자가 질의를 작성할 때 사용한 온톨로지에 정의되어 있는 클래스들로만 구성된 클래스 인덱스를 빠르게 찾을 수 있도록 하여 질의와 관련없는 처리를 줄이기 위한 목적을 가지고 있다. 온톨로지 인덱스는 (온톨로지 아이디, 클래스 인덱스 아이디) 구조의 엔트리들로 구성되며 B+-트리로 구현한다. 유비쿼터스 환경에서는 다양한 서비스를 제공하기 때문에 각 서비스에 대한 의미를 정확하게 기술하기 위해서는 여러 개의 온톨로지가 정의되어 사용될 수 밖에 없다. 그리고 이러한 유비쿼터스 환경이 보편화되고 온톨로지와 메타데이터의 필요성에 대한 인식이 높아지면서 사용자가 질의 처리를 요청할 때 질의와 관련된 온톨로지를 선택하여 질의 조건으로 제시하는 것이 일반화될 것이기 때문에 온톨로지 인덱스는 반드시 필요하다.



(그림 4) 레지스터 인덱스

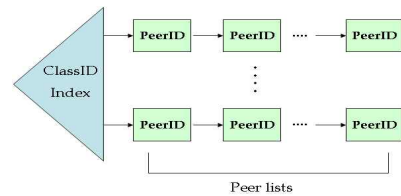
클래스 인덱스는 클래스의 이름을 검색 키로 하여 질의 조건으로 주어진 클래스와 관련된 정보 리소스를 포함하고 있는 클러스터의 코디네이터를 식별하기 위한 아이디와 함께 해당 클래스에 부여된 아이디 정보를 제공한다. 특히 OWL은 같은 의미를 가지고 있지만 이름이 다른 클래스 관계(equivalentClass)를 표현할 수 있기 때문에 본 논문에서는 클래스 인덱스를 구성할 때 이러한 이음동어 관계에 있는 모든 클래스에 대해 같은 아이디를 부여하고 같은 코디네이터가 검색될 수 있도록 한다.

4.3 코디네이터의 데이터 관리

같은 온톨로지의 일부분을 이용하기 때문에 논리적으로 유사한 정보 리소스 또는 서비스를

제공하고 있는 장치들의 클러스터를 관리하는 코디네이터는 온톨로지 레지스터로부터 전달받은 질의를 분석한 후 질의 처리에 가장 적합한 장치를 선별하여 재전달한다. 특히 코디네이터는 온톨로지에 대한 추론 결과에 기반하여 사용자의 질의를 확장해야 하는 중요한 역할을 담당한다. 따라서 온톨로지에 정의된 클래스의 계층 관계와 프로퍼티의 역관계 및 대칭관계에 따라 질의를 확장하기 위한 규칙과 주요 질의 조건으로 제시된 클래스에 관련된 장치의 검색을 지원하기 위한 인덱스 구조가 필요하다.

(그림 5)에서는 클래스의 아이디를 검색 키로 하여 각 클래스에 관련된 장치의 아이디 정보를 유지하고 있는 코디네이터 인덱스의 구조를 보여준다. 코디네이터 인덱스는 크게 클래스의 아이디로 구성된 B+-트리의 클래스 인덱스와 해당 클래스와 관련있는 장치 아이디들로 구성되는 리스트로 구성된다.



(그림 5) 코디네이터 인덱스

코디네이터 인덱스를 이용해서 질의를 확장할 때 적용하는 규칙은 다음과 같다.

1. **클래스의 계층 관계에 기반하여 질의를 확장한다.**
 - 1.1 질의 조건으로 제시된 클래스의 아이디에서 계층 구조의 아이디와 소수 레벨을 추출한다.
 - 1.2 질의 조건과 같은 계층 구조에 있는 클래스들 중에서 제시된 클래스의 하위 클래스로 정의된 것을 모두 검색한다.
 - 1.3 질의 문장에서 원래 질의 조건으로 주어진 클래스를 검색된 하위 클래스로 변경하여 새로운 질의 문장을 생성한다.
2. **프로퍼티의 역관계에 기반하여 질의를 확장한다.**
 - 2.1 역관계 테이블에서 질의 조건으로 주어진 프로퍼티와 역관계가 성립하는 프로

퍼티를 검색한다.

2.2 질의문에서 주어와 목적어의 위치를 서로 바꾸고 프로퍼티를 역관계의 프로퍼티로 변경하여 새로운 질의 문장을 생성한다.

3. 프로퍼티의 대칭관계에 기반하여 질의를 확장한다.

3.1 대칭관계 리스트에 질의 조건으로 주어진 프로퍼티가 존재하는지를 검사하여 해당 프로퍼티에 대칭관계가 존재하는지를 판단한다.

3.2 대칭관계가 존재하는 프로퍼티라면 질의문에서 주어와 목적어의 위치만을 서로 바꾸어 새로운 질의 문장을 생성한다.

코디네이터는 앞서 설명한 확장 규칙을 통해 새로운 질의문들이 생성되면 클래스 인덱스에서 관련된 장치를 검색하여 질의문을 전달한다.

4.4 각 장치의 데이터 관리

각 장치에는 실제로 사용자에게 필요한 정보 리소스에 대한 메타데이터 정보가 저장된다. 따라서 코디네이터로부터 전달된 질의에 관련된 메타데이터 정보를 검색하여 사용자에게 제공하는 것이 각 장치의 가장 중요한 역할이라 할 수 있다. 이러한 역할을 수행하기 위해 본 논문에서는 (그림 6)과 같은 저장 구조를 제안한다.

DomainClass	DomainRID	Property	RangeClass	RangeRID
-------------	-----------	----------	------------	----------

(그림 6) 각 장치의 메타데이터 테이블

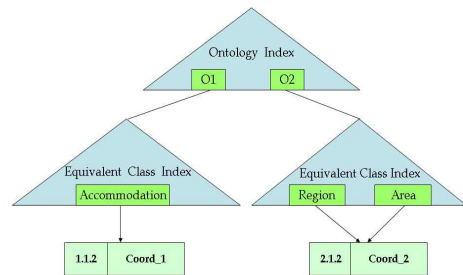
(그림 6)에서 제시한 메타데이터 테이블은 트리플 구조로 기술되는 메타데이터의 구조를 그대로 반영한다는 특징을 가지고 있다. (그림 6)의 테이블에서 주어에 해당하는 정보 리소스의 아이디는 DomainRID 필드에, 해당 리소스의 클래스 타입은 DomainClass 필드에 저장한다. 이와 유사하게 목적어에 해당하는 정보 리소스의 아이디와 클래스 타입은 RangeRID와 RangeClass 필드에 유지하고 주어와 목적어의 프로퍼티 관계는 Property 필드에 저장한다. 질의 문장도 트리플 형태이기 때문에 (그림 6)의 테이블을 이용하면 사용자가 검색하기를 원하는

요소를 쉽게 검색할 수 있는 장점이 있다.

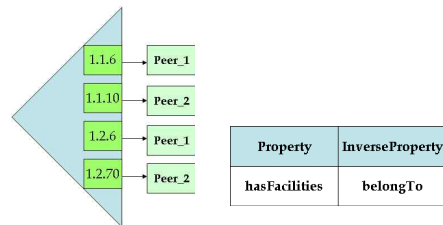
5. 질의 처리 전략

본 절에서는 유비쿼터스 기반의 시맨틱 데이터 관리 시스템에서 사용자 요구에 부합하는 장치를 검색하는 세부 과정을 살펴본다.

(그림 1)에서 제시한 두 개의 온톨로지를 이용하는 것을 가정하고 동일한 온톨로지를 공유하는 장치들을 두 개의 논리적 클러스터로 분류한다. 따라서 첫 번째 코디네이터는 숙박 시설에 대한 온톨로지를 사용하는 두 개의 장치들을 관리하고 두 번째 코디네이터는 관광지에 대한 온톨로지를 사용하는 또 다른 두 개의 장치들을 관리한다고 가정한다.



(그림 7-a) 레지스터 인덱스의 예



(그림 7-b) Coord_1 코디네이터의 인덱스와 역관계 테이블

DomainClass	DomainRID	Property	RangeClass	RangeRID
Hotel	Hotel_1	hasFacilities	Casino	Casino_1
Casino	Casino_1	belongTo	BusinessHotel	BusinessHotel_1

(그림 7-c) Peer_1 장치의 메타데이터 테이블

(그림 7) 시스템 구성 예

(그림 7-a)는 온톨로지 레지스터에 구성되는 인덱스 내용의 일부분을 보여준다. (그림 7-b)는

Coord_1 코디네이터에서 유지하고 있는 인덱스의 일부분과 프로퍼티의 역관계에 대한 정보를 유지하고 있는 역관계 테이블의 일부분을 보여준다. (그림 7-c)는 Peer_1 장치에 존재하는 메타데이터 테이블의 일부분을 보여준다.

만약 사용자가 “Casino_1 리소스를 부대시설로 포함하고 있는 숙박 시설에 대한 정보”를 요구했을 때 다음과 같은 처리 과정을 거쳐서 최종 결과를 반환한다.

▪ **1 단계 : 질의문을 트리플 형태로 변환**

사용자의 질의문은 다음과 같이 표현되며 주요 질의 조건은 “Accommodation” 클래스이다.

1:(Accommodation:? hasFacilities Casino_1)

▪ **2 단계 : 온톨로지 레지스터에서 질의 조건 클래스와 관련된 코디네이터를 검색하여 질의를 전달**

질의는 숙박 시설에 대한 내용이기 때문에 관광지와 관련된 두 번째 온톨로지는 처리 대상에서 제외시킨다. (그림 7-a)에서 제시한 레지스터 인덱스를 이용하여 첫 번째 온톨로지 O1에 연결되어 있는 첫 번째 클래스 인덱스에서 “Accommodation” 클래스의 아이디인 1.1.2와 관련된 코디네이터에 대한 정보를 검색한다. 사용자의 질의는 이제 Coord_1 아이디의 코디네이터로 전달된다.

▪ **3 단계 : 코디네이터에서 질의를 확장하고 관련 장치를 검색하여 질의를 전달**

온톨로지 레지스터로부터 전달받은 클래스 아이디 1.1.2로부터 질의 조건으로 주어진 클래스가 1번 온톨로지에서 1번 계층 구조에 포함되어 있으며 부여된 소수 레이블이 2라는 정보를 분석한다. 분석된 정보를 바탕으로 (그림 7-b)의 인덱스를 이용해서 같은 계층 구조에 포함되어 있으면서 주어진 클래스의 하위 클래스를 검색해보면 아이디가 1.1.6인 “Hotel”과 아이디가 1.1.10인 “Pension”이 반환된다. 4.3절에서 설명한 질의 확장 규칙 1에 따라 다음과 같이 두 개의 질의가 추가된다.

2:(Hotel:? hasFacilities Casino_1)

3:(Pension:? hasFacilities Casino_1)

다음으로 (그림 7-b)에서 제시된 역관계 테이블

을 통해 질의문의 “hasFacilities” 프로퍼티는 “belongTo”와 역관계에 있다는 것을 확인하고 질의 확장 규칙 2번에 의해 다음과 같이 3개의 질의가 추가된다.

4:(Casino_1 belongTo Accommodation:?)

5:(Casino_1 belongTo Hotel:?)

6:(Casino_1 belongTo Pension:?)

(그림 7-b)의 인덱스에서 각 질의와 관련된 장치를 검색하여 해당 질의를 전달한다. 예를 들어 아이디 1.1.6의 “Hotel” 클래스와 관련된 2번과 5번 질의는 Peer_1 장치로 전달되고 아이디 1.1.10의 “Pension” 클래스와 관련된 3번과 6번 질의는 Peer_2 장치로 전달된다.

▪ **4 단계 : 장치에서 각 질의를 처리하여 최종적으로 결과 리소스를 검색**

각 장치에서 질의를 전달받으면 메타데이터 테이블에서 사용자의 요구에 부합하는 정보 리소스를 검색하고 최종 결과로 반환한다. 2번 질의의 경우에는 RangeRID 필드에 해당하는 값이 Casino_1이고 Property 필드에 해당하는 값이 hasFacilities, DomainClass 필드에 해당하는 값은 Hotel이므로 최종적으로 사용자의 요구에 부합하는 정보 리소스는 DomainRID 필드의 값인 “Hotel_1”이 반환된다. 나머지 질의도 같은 방법으로 각 장치에서 처리된다.

6. 결론

언제 어디서나 어떠한 장치를 통해서라도 컴퓨팅이 가능한 것을 목표로 하는 유비쿼터스 환경을 보편화시키기 위해서는 같은 상황이라도 사용자에게 따라 다른 서비스를 제공할 수 있는 사용자 중심의 지능화된 유비쿼터스 컴퓨팅 기반 시스템 개발이 요구된다. 이러한 시스템에는 계속적으로 변화하는 상황 정보를 인지하고 각 사용자의 다양한 요구에 적합한 서비스를 추론할 수 있도록 사용자와 장치들간의 지식 공유와 명확한 의사 소통을 위한 방법이 필요하다. 이를 위해 본 논문에서는 이질적이고 분산된 특성을 가지고 있는 유비쿼터스 환경에 시맨틱 웹 기술을 적용하여 의미적 연관성과 추론에 기반한 사

용자와 장치들간의 상호 작용을 제공하고자 한다. 특히, 본 논문은 각 장치들이 제공하는 데이터의 의미와 데이터들간의 관계를 온톨로지와 메타데이터를 이용하여 기술하고 이를 활용하는 지능적 유비쿼터스 컴퓨팅 환경의 시스템 개발을 위한 새로운 프레임워크를 제시한다.

본 논문에서는 각 장치들이 제공하는 시맨틱 데이터를 OWL로 기술하여 다양한 의미적 관계를 표현하고 이러한 정보를 관리하기 위한 저장 및 인덱스 구조를 각 서브 시스템별로 제안하였다. 그리고 시맨틱 데이터의 주요 요소들을 식별하기 위해 레이블링 기법을 사용하였다. 또한 제안된 저장 및 인덱스 구조를 통해 사용자의 요구에 부합하는 장치를 검색하는 질의 처리 전략을 기술하였다.

본 논문에서 제안한 유비쿼터스 시스템의 시맨틱 데이터 관리 기법은 사용자의 요구에 적합한 정보 리소스나 서비스를 제공하는 장치의 보다 정확하고 빠른 검색을 지원할 것으로 예상되며 향후 실험을 통해 성능 개선 효과를 제시하고자 한다.

참 고 문 헌

[1] W. Keith Edwards and Rebecca E. Grinter, "At home with ubiquitous computing: seven challenges", Ubicomp 2001: Ubiquitous Computing, LNCS 2201, 2001.
 [2] Karen Henriksen, Jadwiga Indulska and Andry Rakotonirainy, "Infrastructure for Pervasive Computing : Challenges", Workshop on Pervasive Computing INFORMATIK 01, 2001.
 [3] Berners-Lee, James Hendler, and Ora Lassila, The Semantic Web, Scientific American, May 2001.
 [4] W3C, OWL Web Ontology Language Guide, <http://www.w3.org/TR/2004/REC-owl-guide-2004-0210/>, 2004.
 [5] Wolfgang Nejdl, Martin Wolpers, Wolf Siberski, Christoph Schmitz, "Super-Peer Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks", In Proc. of the WWW2003, 2003.
 [6] Knarig Arabshian and Henning Schulzrinne, "Combining Ontology Queries with Key Word Search in the GloServ Service Discovery System", In Proc. of the 8th ACM/IFIP/USENIX international conference on Middleware, 2007.

[7] Tharaka Devadithya and Kenneth Chiu, "Index Structure for efficient querying of Distributed Triplestores", In Proc. of the 3rd IEEE International Conference on e-Science and Grid Computing, 2007.
 [8] Weisong Chen, Cho-Li Wang, and Francis C.M. Lau, "A Collaborative and Semantic Data Management Framework for Ubiquitous Computing Environment", In Proc. of the International Conference of Embedded and Ubiquitous Computing (2004), 2004.
 [9] Sonia Ben Mokhtar, Anupam Kaul, Nikolaos Georgantas and Valérie Issarny, "Efficient Semantic Service Discovery in Pervasive Computing Environments", Middleware 2006, LNCS 4290, 2006.
 [10] 김선영, 권동섭, 이석호, "소수 레이블을 이용한 RDF/S 인덱스 구조", 한국컴퓨터종합학술대회 논문집, 32권 1호, 2005



김 연 희

2000년 : 홍익대학교 컴퓨터공학과 졸업(학사)
 2002년 : 홍익대학교 컴퓨터공학과 대학원 졸업(석사)
 2006년 : 홍익대학교 컴퓨터공학과 대학원 졸업(박사)

2007년~현재 : 부천대학 e-비즈니스과 강의전담교수
 관심분야 : 시맨틱 웹, XML, 분산 데이터베이스, 모바일 데이터베이스



김 지 현

1978년 : 이화여자대학교 수학과 학사
 1994년 : 단국대학교 전자 정보 전공 경영학석사
 2004년 : 단국대학교 진산통계학과 이학박사
 1997년~ : 정보관리 기술사

1998년 ~ 현재 : 서일대학 소프트웨어과 부교수
 관심분야 : 웹 공학, 데이터베이스, 품질 관리